NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS & TELECOMMUNICATIONS
GRADUATE PROGRAM
INFORMATION AND DATA MANAGEMENT

MSc THESIS

Recognizing the Structure and Elements of Contracts Using Word Embeddings

Ilias I. Chalkidis

Supervisors: Manolis Koubarakis, Professor
Ion Androutsopoulos, Associate Professor

ATHENS

NOVEMBER 2016

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΔΙΑΧΕΙΡΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΔΕΔΟΜΕΝΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αναγνώριση της Δομής και Στοιχείων Συμβολαίων με Χρήση Διανυσματικών Παραστάσεων Λέξεων

Ηλίας Ι. Χαλκίδης

Επιβλέποντες: Μανόλης Κουμπαράκης, Καθηγητής
Ίων Ανδρουτσόπουλος, Αναπληρωτής Καθηγητής

ΑΘΗΝΑ

ΝΟΕΜΒΡΙΟΣ 2016

MSc THESIS

Recognizing the Structure and Elements of Contracts Using Word Embeddings

Ilias I. Chalkidis
A.M: M1395

SUPERVISORS: Manolis Koubarakis, Professor
Ion Androutsopoulos, Associate Professor

EXAMINATION COMMITTEE: Dimitrios Gounopoulos, Professor

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αναγνώριση της Δομής και Στοιχείων Συμβολαίων με Χρήση Διανυσματικών Παραστάσεων Λέξεων

Ηλίας Ι. Χαλκίδης
Α.Μ: M1395

ΕΠΙΒΛΕΠΟΝΤΕΣ: Μανόλης Κουμπαράκης, Καθηγητής
Ίων Ανδρουτσόπουλος, Αναπληρωτής Καθηγητής

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: Δημήτριος Γουνόπουλος, Καθηγητής

ABSTRACT

Contracts govern business relationships around the world. There is a growing market of people processing contracts every day during a wide range of tasks chasing both business goals and the necessary legal compliance. Through Natural Language Processing (NLP), we can offer solutions by reengineering contracts' plain text as valuable structured data. The Objective of this thesis is to research and propose a baseline approach for the recognition-extraction of contracts' structure and basic elements. For this purpose we rely on some state of the art language modeling techniques such as word embeddings. We presume that the use of word embeddings will give extra reliability against both hand-crafted feature learning and rule-based approaches. One of our main intentions is also that our system (model) will be capable to operate in real-case scenarios, so we evaluate the need for extensive post-processing and orchestration of the discrete components, which are implemented through a divide-and-conquer fashion. This perspective brings a high promise for both savings (i.e. cost, time, human effort) and quality of service by moving complicated tasks which need reiterative human assistance into reliable automated processes.

SUBJECT AREA: Natural Language Processing, Machine Learning, Software Engineering

KEYWORDS: Entity Extraction, Word Embeddings, Contracts

ΠΕΡΙΛΗΨΗ

Τα συμβόλαια διέπουν τις εταιρικές σχέσεις σε όλο τον κόσμο. Βάση αυτής της αναπτυσσόμενης αγοράς αποτελούν οι επαγγελματίες που διαχειρίζονται συμβόλαια καθημερινά στα πλάισια μιας ευρείας γκάμας εργασιών προσδοκώντας τόσο τους εταιρικούς στόχους όσο και την απαραίτητη νομική συμμόρφωση. Μέσω του πεδίου της Επεξεργασίας Φυσικής Γλώσσας μπορούμε να προσφέρουμε σπουδαίες λύσεις σε αυτό το τομέα, αναδιοργανώνοντας το απλό κειμένο των συμβολαίων σε πολύτιμα δομημένα δεδομένα. Ο σκοπός της παρούσας εργασίας είναι η διερεύνηση και πρόταση μια βασικής προσέγγισης για την αναγνώριση - εξαγωγή της δομής και των βασικών στοιχείων των συμβολαίων. Για το σκοπό αυτό βασιζόμαστε σε μερικές από τις πλέον προηγμένες τεχνικές γλωσσικής μοντελοποίησης όπως οι Διανυσματικές Παραστάσεις Λέξεων. Υποθέτουμε ότι η χρήση των Διανυσματικών Παραστάσεων Λέξεων θα δώσει επιπλέον αξιοπιστία σε τέτοιους είδους σύστηματα σε αντίθεση με προσεγγίσεις που κάνουν χρήση ”χειροποίητα” γνωρισμάτων ή/και συστήματα βασισμένα σε κανόνες. Είναι επίσης βασικός στόχος μας, το συγκεκριμένο σύστημα να δύναται να λειτουργήσει σε πραγματικές εφαρμογές. Για αυτό σκοπό αξιολογούμε την χρήση εκτενούς μετα-επεξεργασίας και ”ενορχήστρωσης” των επιμέρους τμημάτων, που προκύπτουν μέσα από μια διαδικασία ”διαίρει και βασίλευε”. Αυτή η προοπτική δίνει υψηλές προσδοκίες τόσο για την εξοικονόμηση πόρων (λ.χ. κόστος, χρόνος, ανθρώπινη προσπάθεια) όσο και για την ποιότητα των υπηρεσίων μετατρέποντας πολύπλοκες εργασίες, που χρίζουν επαναλαμβανόμενης ανθρώπινης διαχείρισης σε αξιόπιστες αυτοματοποιημένες διεργασίες.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Επεξεργασία Φυσικής Γλώσσας, Μηχανική Μάθηση, Τεχνολογία Λογισμικού

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Εξαγωγή Οντοτήτων, Διανυσματικές Παραστάσεις Λέξεων, Συμβόλαια

ACKNOWLEDGMENTS

First of all I would like to thank my supervisor Prof. Ion Androutsopoulos for giving me the opportunity to work on this challenging task. I would also like to praise his mentoring through the whole process and in the same time thank all the members of AUEB's NLP group for their support. Secondly I would like to acknowledge my co-supervisor Prof. Manolis Koubarakis, who supported my efforts through my whole graduate studies and research. Also, I would like to express my gratidute to both Achilleas Michos and Vasilis Tsolis, who trusted me and supported me. It was a pleasure to work and cooperate with all of them. In the 21st century every computer scientist should acknowledge the online community, which provide all of us with a massive amount of knowledge and solutions. Last but not least I would like to thank my family, my parents and my big brother, who support me unconditionally all these years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Contracts are the legal form of agreements with specific terms between two or more parties[2]. These contracting parties mutually agree to sign and validate such a contract to arrange their obligations (corporate or any other kind of obligation). Law firms, legal departments of corporations, governmental bodies all over the world track every day those contracts to extract the appropriate information for a wide range of tasks; chasing both business goals and the necessary legal compliance. The interested parties have high expectations to digitalize their portfolio (archive) and handle it in a structured way through ERP (Enterprise Resource Planning) systems. In such systems there are also demands for structured quering in order to extract valuable information regarding:

- contracts related to specific contracting parties and time periods;
- expiration date of specific contracts, minding the renewal of them or even automated notifications for this puprose;
- other legal information as the referenced legislation or the applicable law and the enforced jurisdiction.

There are hundreds of different tasks that need to be tackled, so the stakeholders need the appropriate tools to overcome the vast workload. For this cause in this thesis we propose a baseline approach for the extraction of contracts' structure and basic elements. Through Natural Language Processing (NLP), we can offer solutions by reengineering contracts' plain text into valuable structured data, applying state of the art language modeling techniques, such as word embeddings, to build these tools. This perspective brings a high promise for both savings (i.e. cost, time, human effort) and quality of service by turning complicated tasks which need reiterative human assistance into reliable automated processes.

1.1 Objective of this thesis

The objective of this thesis is to research and propose a baseline approach for the extraction of contracts' structure and basic elements. For this purpose we rely on some state of the art language modeling techniques such as word embeddings. We presume that the use of word embeddings will give extra reliability against both hand-crafted feature learning and rule-based approaches. One of our fundamental purposes is also that our system (model) will be capable to operate in real-case scenarios, so we evaluate the need for extensive post-processing and orchestration of the discrete components, inherited by a divide-and-conquer fashion.

[2]Parties are considered to be legal persons, which may be a private (i.e., business entity or non-governmental organisation) or public (i.e., government) organisation, and natural persons (individuals).

1.2 Structure for the remaining of the thesis

The next chapter (Chapter 2) presents the related work of this thesis. In Chapter 3 we introduce the appropriate background on contracts' structure and content, in order to be familiarised with the shape of the data we are handling. On Chapter 4 we give a background on the technical aspects of this work which include both data representation and well-known learning algorithms we rely on for the experiments and the metrics for results' evaluation. After that (Chapter 5), it is important to explain the pipeline of the appropriate tasks for the implementation of such a challenging system. Next is Chapter 6, the core of our research dealing with the preparation of data, the experiments and the evaluation of the results. In the last chapter (Chapter 7) we present the conclusion of this research work and discuss future work.

2. RELATED WORK

To the best of our knowledge, limited research has been published regarding the recognition of contracts' structure and elements, with the exception of the article of M.Curtotti and E.Mccreath [1], discussing text classification on a corpus of Australian contracts. In this work 32 types of entities such as dates, parties, clauses etc were classified in a *multi-class* fashion using both machine learning and hand-coded rules. The classification was performed *per line* (sentence) which is by definition less specific in contrast to our own work, in which occasion the classification is performed *per token* (word) and also *per category* to provide more concrete - specific and structured knowledge. Althought the article is highly related to our own work, the results are presented on multi-class form, which excludes a possible comparison.

Another paper, relevant to our field of research discussing named entity recognition in legal text comes from C.Dozier et al., 2010 [2], which explores named entity recognition and resolution in other forms of legal documents such as US case law, depositions, pleadings and other trial documents applying lookup, context rules and statistical models. The types of the entities include titles, companies, jurisdictions, and courts. Except title, the rest of the entities were examined exclusively with lookup and context rules. As authors mention, lookup and rules lack in terms of generalization (e.g. recognize unidentified companies or judicial bodies) as they rely on static lists and hand-crafted work.

We also have to mention the article of P.Quaresma and T.Gonçalves [3], which among other issues, discusses recognition of entities such as locations, organizations, dates and document references in a corpus of legal documents from the Eur-lex site. This article lacks more than others, which we already mentioned, the concernment of distinction between highly correlated entities (i.e. Start Date vs Effective Date vs Termination Date or Governing Law vs Jurisdiction or Contracting Parties vs any other referred party)), which are dependent to the context, an aspect that has been a great challenge in our own work. Also, none of the related publications apply word embeddings as feature representation, which we think they act as the key factor to resolve both the ambiguity between correlated entities, while stacked in sliding windows capture context semantics and achieve the respected generalization.

Both previous articles are published in the book of Francesconi et al. [4] among others, also discussing information retrieval on legal texts.

There are various articles discussing extensively the task of general purpose entity extraction using word embeddings [5] [6] [7].

3. BACKGROUND ON CONTRACTS

Most of us are not really aware of the form and the textual information of the contracts, so it is our first concern to get familiarised with the contracts and understand the shape of the data. The text of contracts is full of valuable information. In this work we are considering the core of this information: the structure of the clauses and the main contract elements which are mentioned in the majority of any type of contract.

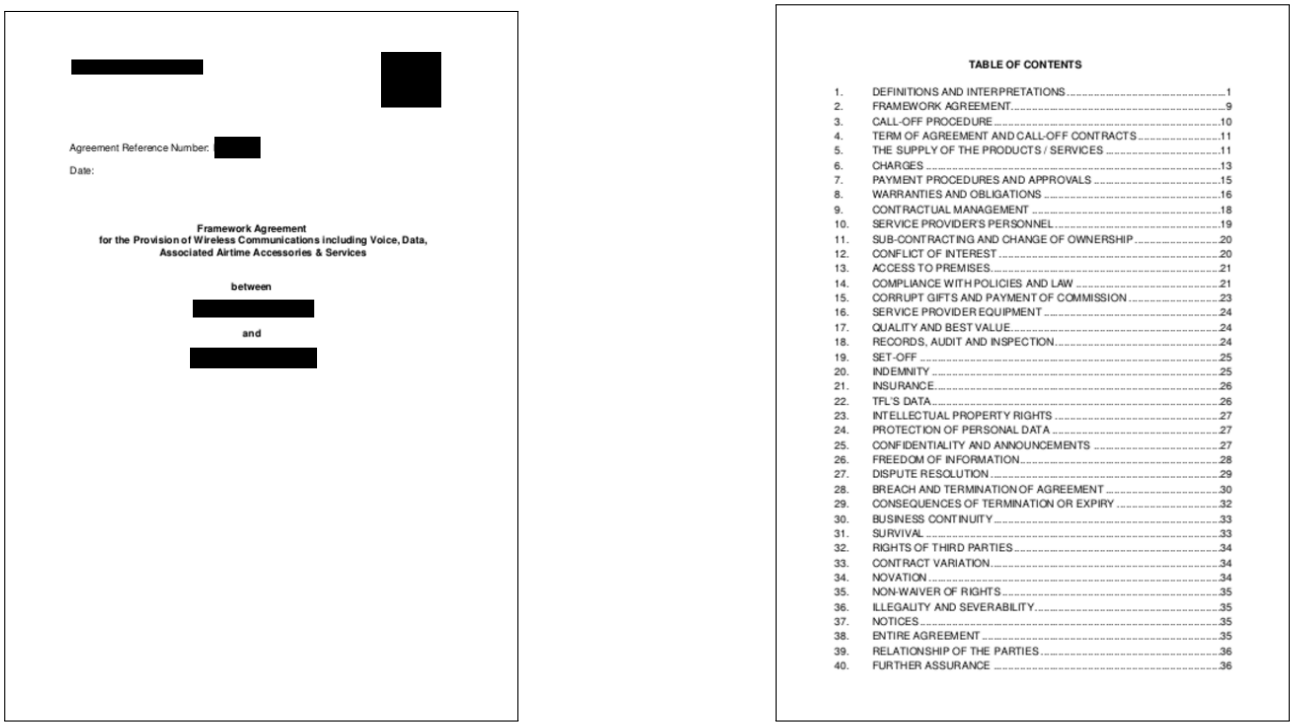
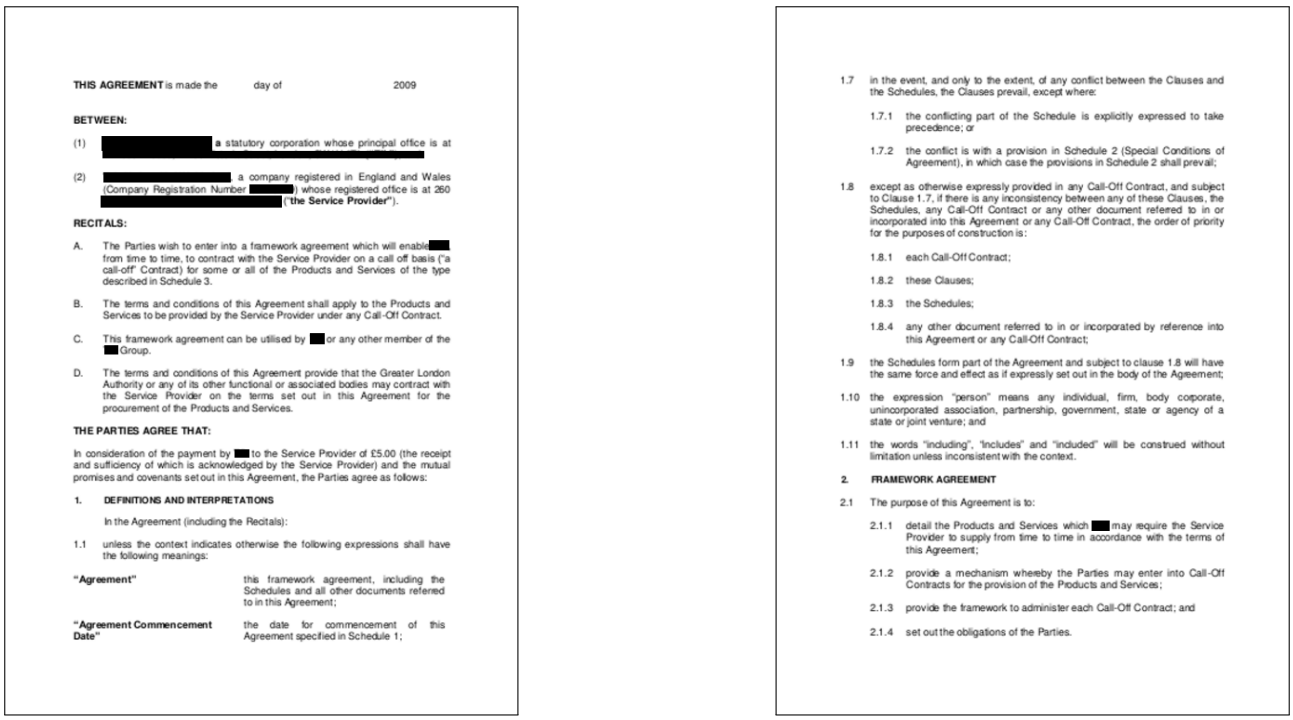


Figure 3.1: Contract Template: Cover page and Table of Contents

3.1 Contract Structure

Contracts adhere to a typical structure starting with an introduction / preamble with the contract title, the contracting parties entering into the agreement and reference to the start or effective date of the contract. It is a common practice to use a cover page with the same information followed by a table of contents (See Figure 3.1). The introduction is usually followed by the recitals which provide us with some background to the agreement and afterwards the main body of the contract which includes the terms of the contract and is organised in clauses (See Figure 3.2).



THIS AGREEMENT is made the day of 2009

BETWEEN:

(1) [redacted] a statutory corporation whose principal office is at [redacted]

(2) [redacted] a company registered in England and Wales (Company Registration Number [redacted]) whose registered office is at 260 [redacted] ("**the Service Provider**").

RECITALS:

A. The Parties wish to enter into a framework agreement which will enable [redacted] from time to time, to contract with the Service Provider on a call off basis ("a call-off" Contract) for some or all of the Products and Services of the type described in Schedule 3.

B. The terms and conditions of this Agreement shall apply to the Products and Services to be provided by the Service Provider under any Call-Off Contract.

C. This framework agreement can be utilised by [redacted] or any other member of the [redacted] Group.

D. The terms and conditions of this Agreement provide that the Greater London Authority or any of its other functional or associated bodies may contract with the Service Provider on the terms set out in this Agreement for the procurement of the Products and Services.

THE PARTIES AGREE THAT:

In consideration of the payment by [redacted] to the Service Provider of £5.00 (the receipt and sufficiency of which is acknowledged by the Service Provider) and the mutual promises and covenants set out in this Agreement, the Parties agree as follows:

1. **DEFINITIONS AND INTERPRETATIONS**

In the Agreement (including the Recitals):

1.1 unless the context indicates otherwise the following expressions shall have the following meanings:

"Agreement"	this framework agreement, including the Schedules and all other documents referred to in this Agreement;
"Agreement Commencement Date"	the date for commencement of this Agreement specified in Schedule 1;

1.7 in the event, and only to the extent, of any conflict between the Clauses and the Schedules, the Clauses prevail, except where:

1.7.1 the conflicting part of the Schedule is explicitly expressed to take precedence; or

1.7.2 the conflict is with a provision in Schedule 2 (Special Conditions of Agreement), in which case the provisions in Schedule 2 shall prevail;

1.8 except as otherwise expressly provided in any Call-Off Contract, and subject to Clause 1.7, if there is any inconsistency between any of these Clauses, the Schedules, any Call-Off Contract or any other document referred to in or incorporated into this Agreement or any Call-Off Contract, the order of priority for the purposes of construction is:

1.8.1 each Call-Off Contract;

1.8.2 these Clauses;

1.8.3 the Schedules;

1.8.4 any other document referred to in or incorporated by reference into this Agreement or any Call-Off Contract;

1.9 the Schedules form part of the Agreement and subject to clause 1.8 will have the same force and effect as if expressly set out in the body of the Agreement;

1.10 the expression "person" means any individual, firm, body corporate, unincorporated association, partnership, government, state or agency of a state or joint venture; and

1.11 the words "including", "includes" and "included" will be construed without limitation unless inconsistent with the context.

2. **FRAMEWORK AGREEMENT**

2.1 The purpose of this Agreement is to:

2.1.1 detail the Products and Services which [redacted] may require the Service Provider to supply from time to time in accordance with the terms of this Agreement;

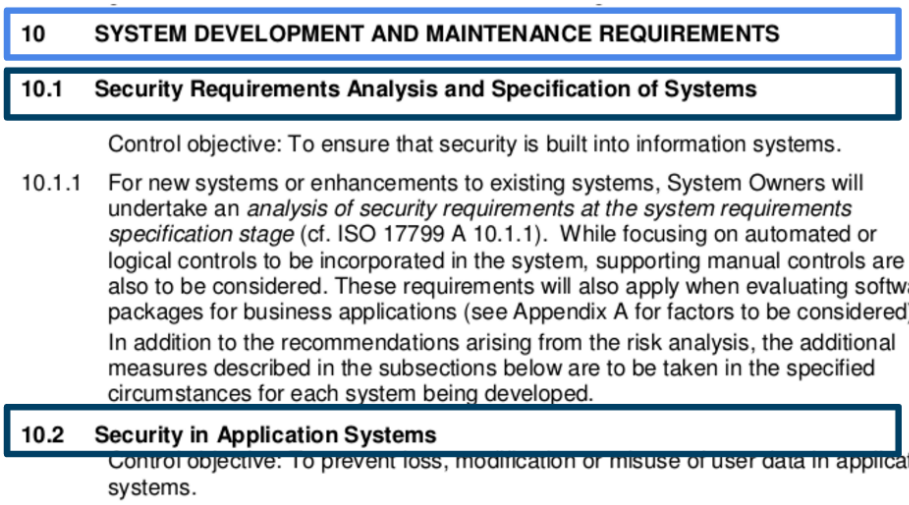
2.1.2 provide a mechanism whereby the Parties may enter into Call-Off Contracts for the provision of the Products and Services;

2.1.3 provide the framework to administer each Call-Off Contract; and

2.1.4 set out the obligations of the Parties.

Figure 3.2: Contract Template: Introduction - Recitals and Main Body (Clauses)

Clauses are the structural units of the contract's text and they are hierarchically structured in parts, chapters, sections and other forms of numbered paragraphs. Numbering and heading are needed for a paragraph to be considered as a named clause and be part of the main body structure (See Figure 3.3).



10 SYSTEM DEVELOPMENT AND MAINTENANCE REQUIREMENTS

10.1 Security Requirements Analysis and Specification of Systems

Control objective: To ensure that security is built into information systems.

10.1.1 For new systems or enhancements to existing systems, System Owners will undertake an *analysis of security requirements at the system requirements specification stage* (cf. ISO 17799 A 10.1.1). While focusing on automated or logical controls to be incorporated in the system, supporting manual controls are also to be considered. These requirements will also apply when evaluating software packages for business applications (see Appendix A for factors to be considered).

In addition to the recommendations arising from the risk analysis, the additional measures described in the subsections below are to be taken in the specified circumstances for each system being developed.

10.2 Security in Application Systems

Control objective: To prevent loss, modification or misuse of user data in application systems.

Figure 3.3: Contract Template: Main Body

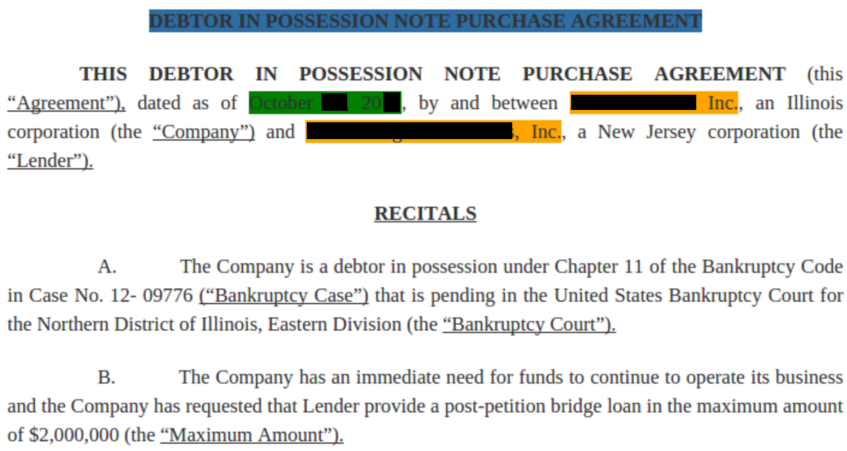
3.2 Contract Elements

By the naming of Contract Elements, we refer to the following entities (See Figures 3.4 - 3.8):

Contract Title / Type includes information about the type of the contract (e.g. employment, securities, loan etc) and versional information in cases of amendments. It is usually written on the front of the cover and the first page of the contract.

Contracting Parties are the named legal and natural persons (entities) which accept the benefits and obligations specified on the contract. They are usually written onto the front of the cover and the first page of the contract.

Start Date is generally the date that the last party signed the contract. This date is usually the date which both parties consider to be the date the contract was made and became effective, unless there is a different defined *Effective Date*. It is usually written onto the front of the cover and the first page of the contract.



DEBTOR IN POSSESSION NOTE PURCHASE AGREEMENT

THIS DEBTOR IN POSSESSION NOTE PURCHASE AGREEMENT (this "Agreement"), dated as of October [redacted] 20[redacted], by and between [redacted] Inc., an Illinois corporation (the "Company") and [redacted], Inc., a New Jersey corporation (the "Lender").

RECITALS

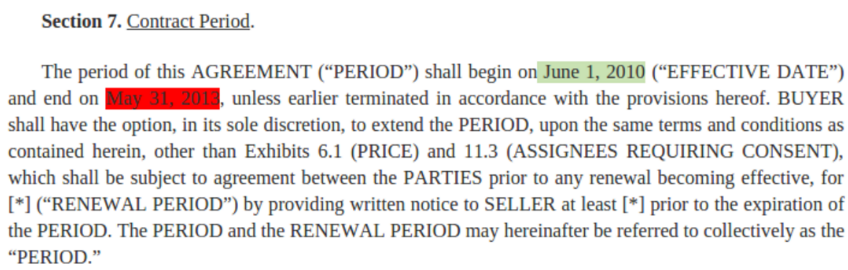
A. The Company is a debtor in possession under Chapter 11 of the Bankruptcy Code in Case No. 12- 09776 ("Bankruptcy Case") that is pending in the United States Bankruptcy Court for the Northern District of Illinois, Eastern Division (the "Bankruptcy Court").

B. The Company has an immediate need for funds to continue to operate its business and the Company has requested that Lender provide a post-petition bridge loan in the maximum amount of $2,000,000 (the "Maximum Amount").

Figure 3.4: Highlighted Contract Elements: Title - Contracting Parties - Start Date

Effective Date is the date that contracting parties consider the contract effective. Parties may be in negotiations for months before the date of the contract and then refer to the date they started negotiations as being the effective date. It is usually written onto the first page of the contract or the definitions clause.

Termination Date is the date in which the contract is terminated. It is usually written onto the definitions or termination (term) clause.



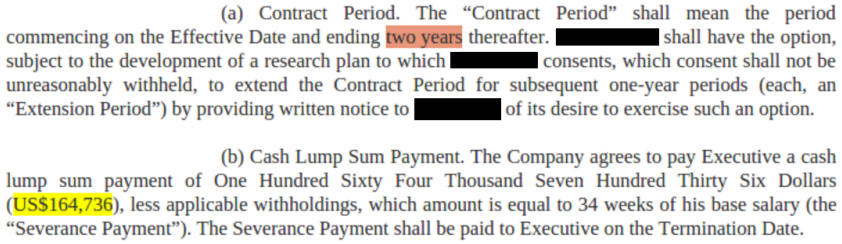
Section 7. Contract Period.

The period of this AGREEMENT ("PERIOD") shall begin on June 1, 2010 ("EFFECTIVE DATE") and end on May 31, 2013, unless earlier terminated in accordance with the provisions hereof. BUYER shall have the option, in its sole discretion, to extend the PERIOD, upon the same terms and conditions as contained herein, other than Exhibits 6.1 (PRICE) and 11.3 (ASSIGNEES REQUIRING CONSENT), which shall be subject to agreement between the PARTIES prior to any renewal becoming effective, for [*] ("RENEWAL PERIOD") by providing written notice to SELLER at least [*] prior to the expiration of the PERIOD. The PERIOD and the RENEWAL PERIOD may hereinafter be referred to collectively as the "PERIOD."

Figure 3.5: Highlighted Contract Elements: Effective Date - Termination Date

Contract Period specifies the number of working or calendar days, from a specified effective date to a specified completion (termination) data, as provided for in a contract. It is usually written onto the definitions or termination (term) clause.

Contract Value is the negotiated or proposed price of a contract. It is usually written onto the Lump Sum or Cash Considerations clause.



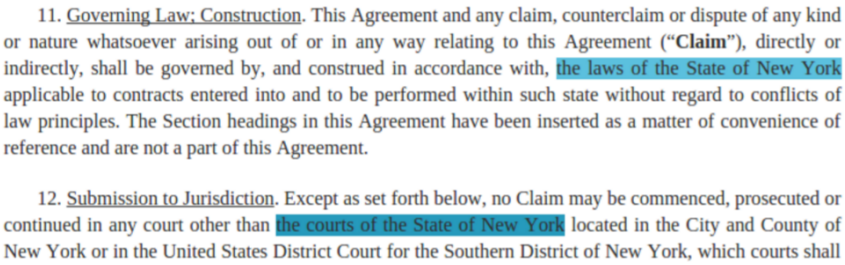
(a) Contract Period. The "Contract Period" shall mean the period commencing on the Effective Date and ending two years thereafter. [redacted] shall have the option, subject to the development of a research plan to which [redacted] consents, which consent shall not be unreasonably withheld, to extend the Contract Period for subsequent one-year periods (each, an "Extension Period") by providing written notice to [redacted] of its desire to exercise such an option.

(b) Cash Lump Sum Payment. The Company agrees to pay Executive a cash lump sum payment of One Hundred Sixty Four Thousand Seven Hundred Thirty Six Dollars (US$164,736), less applicable withholdings, which amount is equal to 34 weeks of his base salary (the "Severance Payment"). The Severance Payment shall be paid to Executive on the Termination Date.

Figure 3.6: Highlighted Contract Elements: Contract Period and Value

Governing Law specifies that the laws of a mutually agreed upon jurisdiction will govern the interpretation and enforcement of the terms of the contract. It is usually written onto the related clause (Governing Law).

Jurisdiction specifies the courts of a named country taking jurisdiction over any disputes that may arise between the contracting parties. It is usually written onto the related clauses (Governing Law, Jurisdiction).

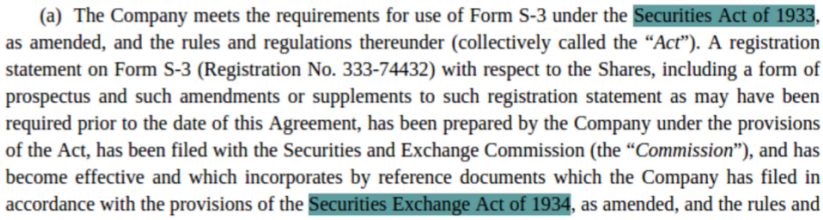


11. Governing Law; Construction. This Agreement and any claim, counterclaim or dispute of any kind or nature whatsoever arising out of or in any way relating to this Agreement ("**Claim**"), directly or indirectly, shall be governed by, and construed in accordance with, the laws of the State of New York applicable to contracts entered into and to be performed within such state without regard to conflicts of law principles. The Section headings in this Agreement have been inserted as a matter of convenience of reference and are not a part of this Agreement.

12. Submission to Jurisdiction. Except as set forth below, no Claim may be commenced, prosecuted or continued in any court other than the courts of the State of New York located in the City and County of New York or in the United States District Court for the Southern District of New York, which courts shall

Figure 3.7: Highlighted Contract Elements: Governing Law - Jurisdiction

Legislation References are the references in any national, community or international law. The references are all over the contract's text.



(a) The Company meets the requirements for use of Form S-3 under the Securities Act of 1933, as amended, and the rules and regulations thereunder (collectively called the "*Act*"). A registration statement on Form S-3 (Registration No. 333-74432) with respect to the Shares, including a form of prospectus and such amendments or supplements to such registration statement as may have been required prior to the date of this Agreement, has been prepared by the Company under the provisions of the Act, has been filed with the Securities and Exchange Commission (the "*Commission*"), and has become effective and which incorporates by reference documents which the Company has filed in accordance with the provisions of the Securities Exchange Act of 1934, as amended, and the rules and

Figure 3.8: Highlighted Contract Elements: Legislation References

4. TECHNICAL BACKGROUND

In this chapter firstly, we will discuss the task of information extraction and more specifically the NLP subtask of entity extraction. In every classification task one of the most crucial decisions is the feature selection. Most named-entity recognition (NER) systems rely on hand-crafted features and on the output of other NLP tasks such as part-of-speech (POS) tagging and text chunking. We will demonstrate the use of word embedding as an alternative robust feature representation and the concept of sliding windows of tokens (words). Last but not least we will mention the classification algorithms and the evaluation metrics we use for experimentation.

4.1 Information Extraction

Information Extraction is the task of automatically extracting structured information from unstructured and/or semi-structured documents. Moreover our work complies with the subtask of entity extraction, also known as Named-Entity Recognition (NER) using NLP techniques over machine learning. Entity extraction is a pure classification problem, where entities correspond to specific categories.

In Natural Language Processing there are two main approaches for entity extraction: sequence tagging and bootstrapping. In sequence tagging, labels (tags) are assigned to each token (word) of training sequences (e.g., training sentences) and then using some learning algorithms the system learns to assign labels to the tokens of unseen (test) sequences. In the second approach, bootstrapping, given some seeds (e.g., known names of person), we collect contexts around the seeds. Then we use the contexts to identify new entity names (e.g., additional person names) and generate additional samples. In our work, there are no evidence that the second approach is also reliable. For example given a specific date from an initial dataset, there is no correlation that any other date in other contracts could be labeled as a start date. The same applies correspondingly to any other category.

4.2 Word embeddings

By the term word embedding in natural language processing (NLP), we describe a feature representation where words (tokens) or phrases (multi-token) from the vocabulary are mapped to vectors of real numbers. Distributed representations of words in a vector space help learning algorithms to achieve better performance in natural language processing tasks by grouping similar words. The ultimate leverage of such a technique is the transition from the traditional sparse features (i.e. one-hot vector representation) onto the dense vector space of common shared features.

4.2.1 Word2vec model

Word2vec is a group of related models that are used to produce word embeddings. These models are two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as input a large corpus of text and produces a high-dimensional space (typically of several hundred dimensions), with each unique word in the corpus being

assigned to a corresponding dense vector in the space. Word vectors are positioned in the vector space in a fashion that enables words that share common contexts in the corpus to be located in close proximity to one another in that space. [8]

Word2vec can utilise either of these two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) or continuous skip-gram (See Figure 4.1). In the training task of the continuous bag-of-words architecture, the model predicts the current word from a window of surrounding context words. In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. According to the authors' note, CBOW is faster while skip-gram is slower but more efficient for infrequent words. [8] [9]

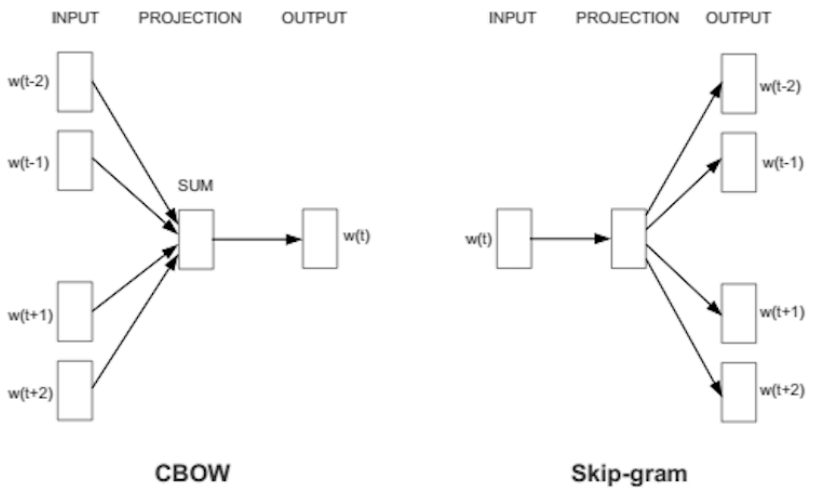


Figure 4.1: The model architectures of word2vec: CBOW and Skip-gram

Given a large corpus of documents, Word2vec can accurately guess a word's meaning based on its appearances and the respective context of these appearances. Those representations give a word's association with other words (e.g. "king" is to "man" what "queen" is to "woman") and can also be used to find relative words. To familiarise with word2vec model, we share two typical examples. In Figure 4.2, we see a list of words associated with "Sweden" using word2vec, in order of proximity based on cosine similarity. In Figure 4.3, we see projections of countries in relation to their capitals using PCA[1] [10] from the high-dimensional space to the two-dimensional one[2].

4.3 Sliding Window of tokens

In the challenging task of classifying a word (token) into a specific category, surrounding context words seem to be really important in the related sequence of words. As we already showed the word2vec model relies on this information in order to map words in the high-dimensional vector space. A sliding window of tokens (words) is considered as the group

[1] Principal Component Analysis (PCA) is used as a dimensionality reduction technique in machine learning. While its main application is reducing the dimensions of the dataset's instances for memory allocation's reasons, in our case it is used to make possible the projection of word embeddings in the the two-dimensional space.

[2] All figures are reproduced from `https://deeplearning4j.org/word2vec`

Word	Cosine Distance
norway	0,760124
denmark	0,715460
finland	0,620022
switzerland	0,588132
belgium	0,585835
netherlands	0,574631
iceland	0,562368
estonia	0,547621
slovenia	0,531408

Figure 4.2: List of words associated with “Sweden” using Word2vec model

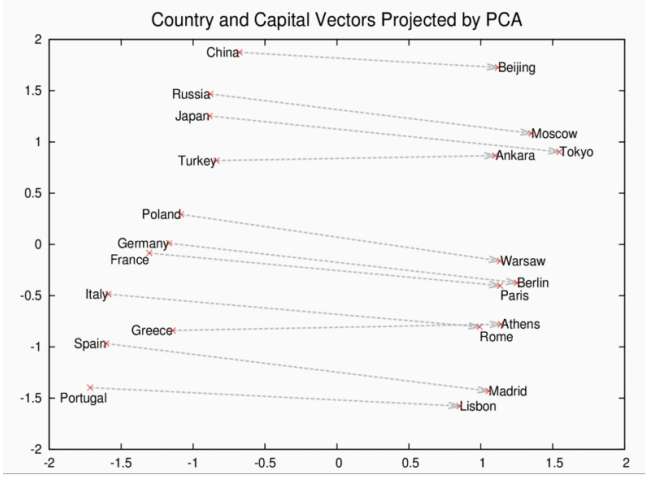


Figure 4.3: Country and Capital Vectors Projected by PCA

of 2k+1 tokens. The group consists of the middle word, which is called the target-examined word, and the k words to each side, which are called the contexts (Figure 4.4). The target - examined word can have several different uses and completely different meaning and purpose according to the surrounding words.

Giving a simple example, a two-digits number like 23 means something completely different between sentence ”I am 23 years old” and the sentence ”I was born in August 23, 1976”. So instead of classifying single words, or single word embeddings, we can classify multiple words (window) or word embeddings in our goal to classify the target word. The size and the shape of these windows may vary according to the complexity of any given problem. Traditional NLP tasks such as part-of-speech (POS) tagging using word embeddings seem to work decently with sliding windows of 3-5 words [11]. This prooved to be highly insufficient in our case, where the contextual information, which is appropriate to distinguish different entities, that seem similar, is extended up to 5 tokens before or after. This fact leads us to the use of sliding windows of 9-11 words.

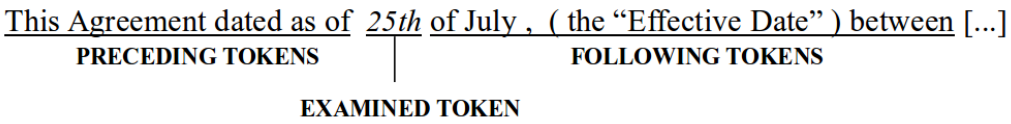


Figure 4.4: Window of tokens

Other well-known techniques used for sequence labeling are: Hidden Markov Models (HMMs) [12] and Recurrent Neural Networks (RNNs) [13]. In this work, we do not look into these techniques which we reserve for future work.

4.4 Classification Algorithms - Evaluation Metrics

4.4.1 Classification Algorithms

In our research, experiments were driven by linear supervised machine learning algorithms: Logistic Regression (LR) [14] and Support Vector Machines (SVM) [15] [16] with linear kernel. Both of these algorithms are widespread and perform decently in many classification problems. Specifically Support Vector Machines (SVM) have exceptional performance in natural language processing (NLP) tasks like Part-of-Speech (POS) tagging, text chunking, among others.

4.4.2 Performance Metrics

Our evaluation is performed under the following metrics:

Precision, the number of successfully retrieved elements (positive) divided by the number of retrieved elements (both positive and negative).

$$Precision = truepositives/(truepositives + falsepositives) \tag{4.1}$$

Recall, the number of successfully retrieved elements (positive) divided by the number of relevant elements (retrieved or not)

$$Recall = truepositives/(truepositives + falsenegatives) \tag{4.2}$$

F-measure, the weighted harmonic mean of precision and recall

$$F - measure = 2 \times (Precision \times Recall)/(Precision + Recall) \tag{4.3}$$

5. THE SYSTEM OF THE THESIS

In this chapter we will extensively analyse the the pipeline of the appropriate tasks for the implementation of such a challenging system. This pipeline begins with the annotation task of contracts, continues with the classification experiments and ends with the post-processing analysis of its results. The final outcome is a matter of the orchestration of all classifiers, enhanced with hand-crafted rule-based processes according to the domain knowledge.

5.1 Annotation Platform

The first important task in every NLP system, missing a public dataset, is the annotation of related documents on which both training and evaluation could rely. There are many available annotations tools like GATE[2] and brat[3] , on which researchers can depend on for their dataset preparation.

Considering the nature of our problem and the desire to control the whole process of annotation, we implemented our own annotation platform (Figure 5.1). By building such a platform using the state-of-art web development technologies (CSS, JS)[4] we could provide annotators a user-friendly environment to work on. In the same time our team had the ability to monitor this process on-the-fly, resolving any further issues as the important task of the final data cleansing.

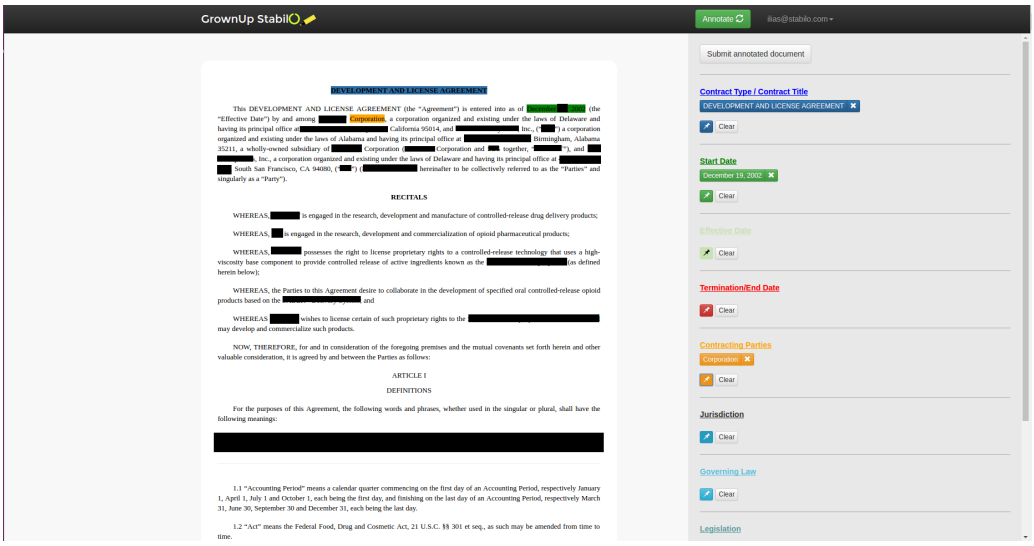


Figure 5.1: Annotation User Interface of the Annotation Platform

[2] https://gate.ac.uk

[3] http://brat.nlplab.org/

[4] Cascading Style Sheets (CSS) and JavaScript (JS) are the main web development technologies which enable HTML to produce a dynamic stylish web user interface.

We want our platform to be user-friendly, as we already mentioned, both in terms of usability and understanding but also in terms of saving time. For this purpose we added a pre-process component in our system to assist users with annotation recommendations.

We decided to deploy two different platforms based on the same architecture. The first platform was deployed for the annotations of all contract elements, and the second one for the annotation of contract structure, meaning named clauses and introduction paragraphs as described in Chapter 3.

The pipeline (Figure 5.2) of our annotation platform can be explained as follows:

Documents Input Contracts should be served in HTML format to help annotators understand the structure of the contract and locate the appropriate information.

Pre-processing - Recommendations As already mentioned, it was really important to have as many annotated contracts as soon as possible. So contracts pass through a pre-process task using regular expressions in some well-defined contract elements, such as dates, or legislation references but also rely on Stanford NER[1] for the recognition of contracting parties.

Annotation The main goal of this pipeline is of course the annotation. Users via a modern Web User Interface could annotate contract elements and contract structure highlighting text areas. The usability of this process was really important to save both time and possible errors, so users had full control to select, deselect or edit their selection and also to quick review their choices.

Save annotated contracts With the final submission, annotated contracts are coming to the server. Documents are saved in two formats: one with the predefined HTML annotation, and the other striped from this HTML annotations, as plain text with only our own annotation tags for each category.

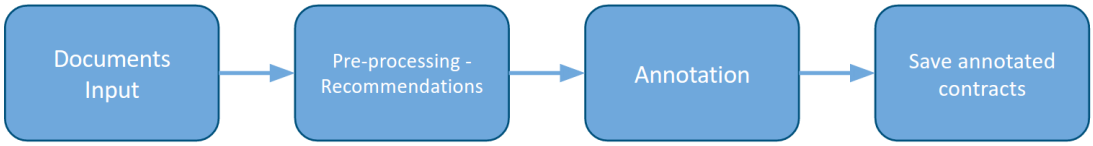


Figure 5.2: Annotation Platform: The Pipeline of the tasks

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

5.2 Classification Experiments

For our classification experiments we follow a traditional approach, well-known in the machine learning field. After the dataset preparation, we split the samples in the proportions of 80/20 in a training set and a test set. A second division (80/20) is appropriate in the training set to have a first evaluation of our model in a validation set. We supply our training algorithms with the samples of the training set and evaluate their performance. Finally we have a stable predefined validation set which is used to calculate the final performance of our models.

All classification experiments are performed in a 2-class fashion, which means that we produce a different IO (In-Out) model for each category (element). An important aspect for the classification experiments is the tuning of the model's hyper-parameters. The work-flow is presented in Figure 5.3.

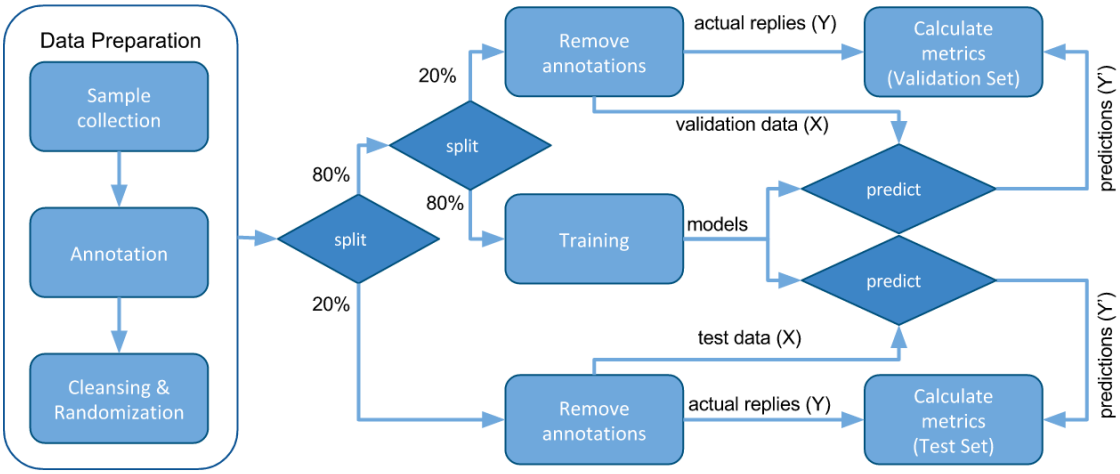


Figure 5.3: Classification Experiments: The pipeline of the tasks

5.3 Post-processing Analysis

Evaluating only the performance of each classifier per token can totally mislead our objective. We want the final outcomes of our system to be multi-token groups (phrases) which comply to the actual elements (entities). For this purpose extensive post-processing is required to sustain the reliability of our system.

The post-processing task can be analysed in the following sub-tasks (See also Figures 5.4-5.5):

Predict The system calls the appropriate classifier to predict the class of input words (tokens).

Group The system create groups of the positive tokens.

Extend The system examine a few words around the group based on hand-crafted rules (e.g. bags of words look-up) in order to find false negative predictions and extend the target group.

Validate The system validates the correctness of the extended groups using hand-crafted rules (e.g. regular expressions).

Post-processings sub-tasks, which rely on hand-crafted rules, vary among different elements to comply with the particularities of each element.

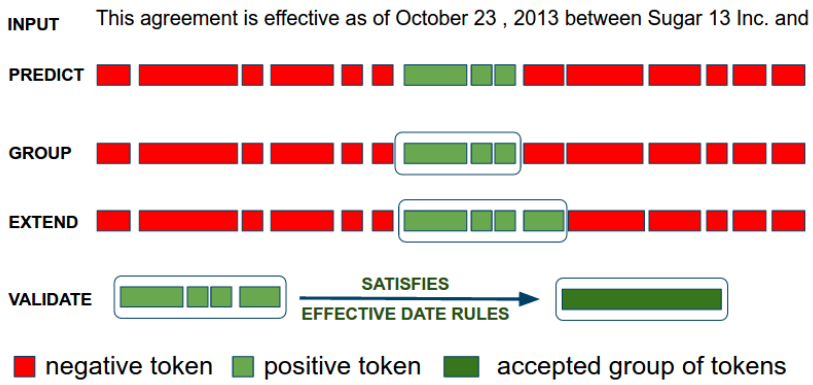


Figure 5.4: Post-processing Analysis: The pipeline of the sub-tasks (Example A)

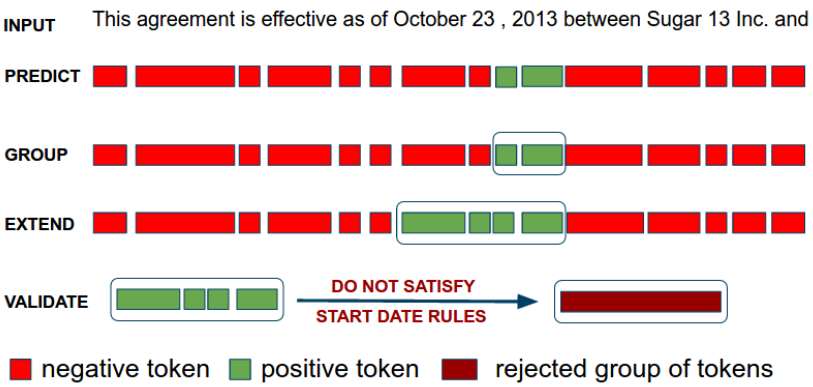


Figure 5.5: Post-processing Analysis: The pipeline of the sub-tasks (Example B)

5.4 System Integration

System integration was one the most challenging tasks. Different components coming from the classification experiments and post-processing have to stack together with some rule-based hand-crafted components. The objective of the individual components' integration is to promote our desire to make our system capable for use in real-case scenarios. This is considered as our proof of concept.

First of all, we have to divide the text in 4 segments, as follows from the analysis that we made in sub-section 3.1: Cover Page, Table of Contents, Introductions / Recitals and Main Body. In this dividing process, we use some rule-based tasks (e.g. recitals recognition), which had no need for advanced machine learning and could be easily resolved with empirical linguistic analysis. In the same section (Section 3), we mention the possible position of each contract element in the contract's text, which we follow accordingly in this orchestration (e.g. the system predicts the Contract Title / Type only in the Cover Page and the Introduction). Finally we use some extra rule-based processing in the definition of specific named clauses, in which some of the contract elements are written. The full orchestration of our system is presented in Figure 5.6.

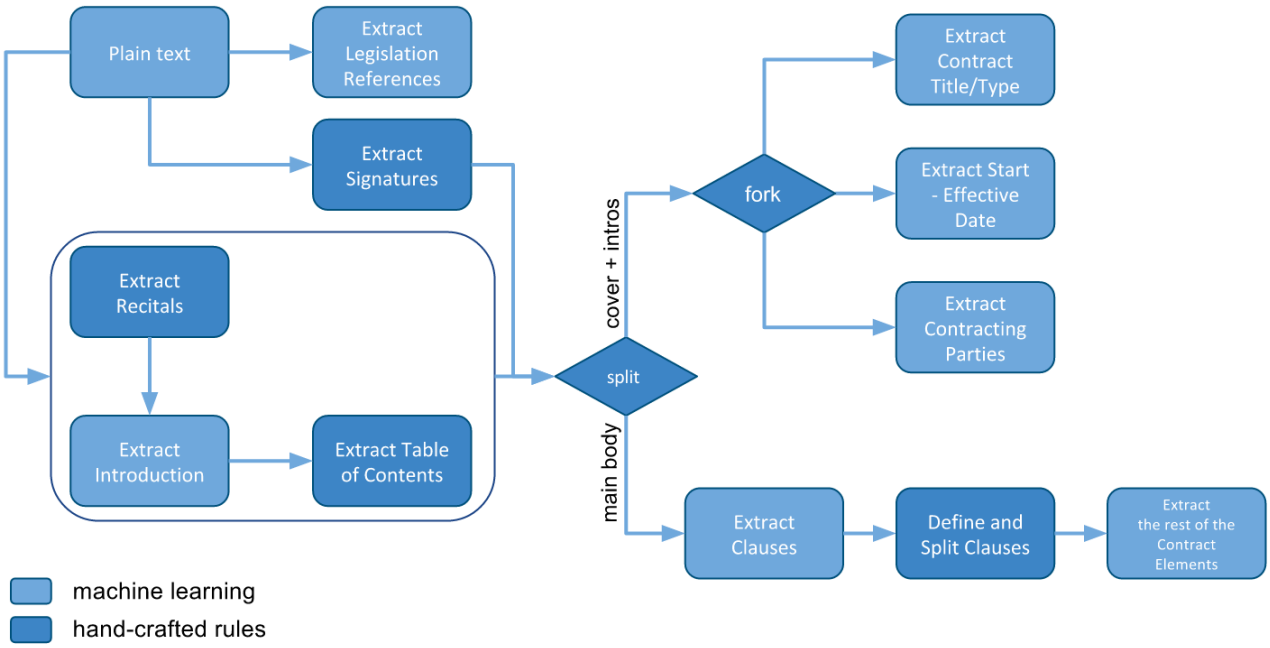


Figure 5.6: System: Orchestration

6. EXPERIMENTS & RESULTS

In this chapter we will extensively present the experiments of our work. We will start with the dataset preparation, meaning the process of building domain-specific word embeddings and the data representation of our samples. Next we will discuss the classification experiments and the corresponding results, with particular reference to the feature representation impact. Finally we will examine the post-processing task and the final results and will compare them with the results of rule-based systems.

6.1 Dataset preparation

6.1.1 Domain-specific word embeddings

Many general purpose pre-trained word embeddings are publicly available. They have been trained with billions of tokens in large corpora. In most occasions these corpora consist of news articles, Wikipedia pages or general pages coming from crawling around the web. Our domain, contracts, on the other hand, is possessed by the legal phraseology, which consists standard phrases (norms) and pretty much consistent vocabulary, which differ from those of a generic language model (news articles, encyclopedia entries, etc). For this reason, we trained our own word2vec model based on a corpus composed strictly of contracts. Our first word2vec model was trained over 36,000 contracts, which sumed up to over 453 million of words (tokens), and finally produced word embeddings for a decent vocabulary of 101,330 words. A great aspect of domain-specific word embeddings is also the ability to handle numbers, special characters and stop words with our own perspective, while in most of the pre-trained models are captured partially. The training was performed using a skip-gram model with the Gensim[2] library [17]. In Table 6.1 we see the similarity we achieved in specific distinctive words. In Figure 6.1, we see the 400 most frequent words projected by the TSNE algorithm. Also in Figure 6.2 with 100 selected significant words to obtain a better perspective of the domain-specific model.

[2] https://radimrehurek.com/gensim/models/word2vec.html

Figure 6.1: Domain-Specific word2vec model: The 400 most frequent word embeddings projected using TSNE

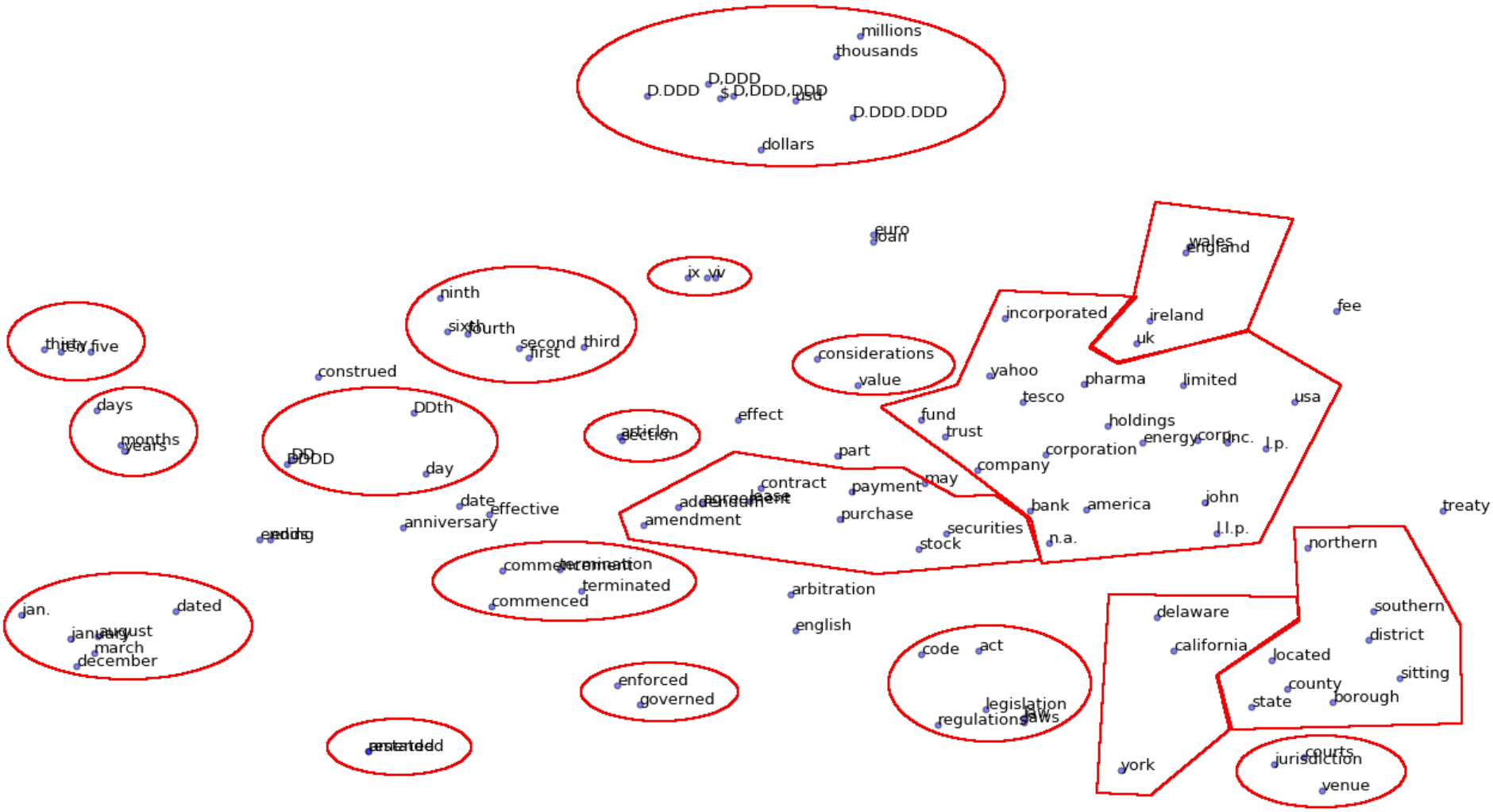


Figure 6.2: Domain-Specific word2vec model: 100 selective significant word embeddings projected using TSNE

Table 6.1: Domain-specific word2vec model: Top-10 similar words

Word	Top-10 similar words
agreement	addendum, agreements, agree/xadment, this, statues, lease, contract, guaranty, sublease, rca, amendment
november	august, april, february, july, october, june, march, september, january, december
inc.	inc, llc, ltd., corp., l.p., l.l.c., inc.., c/o, inc's, d/b/a
laws	law, statutes, legislation, usury, statues, ordinances, statem, antifraud, rules, regulators
court	courts, compenent, judge, tribunal, jurisdiction, body, arbitral, judicial, district, arbitrator
act	sarbones-oxley, puhca, s-k, commission, legislation, s-x, hola, u.s.c, DDDD, irc
article	section, paragraph, clause, miscellanneous, sections, subparagraph, subsection, sub-clause, paragraphs, chapter

6.1.2 Data representation

By performing multiple experiments with word embeddings and hand-crafted features, we ended up using both domain-specific embeddings and hand-crafted features, which differ according to the examined category. For each token we classify, the sample consists of a window of 11 surrounding tokens (words) (the five previous ones, the 5 next and the examined one). For each of these tokens we have the word embedding (200 real numbers) and 15 extra hand-crafted features. In case our word2vec model does not provide us an according word embedding a random uniform of 200 real numbers is assigned. For line breaks we designate a group of 200 zeros. Every sample (token) is represented with a sum of features:

$$11 \times [200(word\_embedding\_size) + 15(hand\_crafted\_features)] = 2365/token. \quad (6.1)$$

We divide the hand-crafted features in two sets: a basic set of 8 features, which are the same for each category, and a set of 7 features, specialised for each distinctive category. The basic features capture information of case sensitivity (all-cased, partially-cased, lowercased), the size of the token, the existence of digits and the role of special characters and stop words. The set of 7 additional features handles common bag of words and regular expressions for each specific category. In most cases, these bag of words refer to common positive words, frequent context around each element and negative words. These bag of words are the outgrowth of our own linguistic analysis, observing our dataset.

6.1.3 Dataset

The annotation task resulted in 2,500 annotated contracts for contract elements' experimentation and 1000 annotated contracts for structure's (clauses and introduction) experimentation. In specific categories we have more samples for two reasons: some of these elements are mentioned in almost every contract (e.g. title, parties, start date etc) and some of these elements are presented multiple times per contract (legislation references and clauses). Some others are rarely mentioned - specified, so in this occasion we lack samples (e.g. termination date, contract period). In Table 6.2, we present statistics per category over the dataset.

Table 6.2: Dataset Analysis: Statistics on the data

Category	percentage	#elements	tokens per element	#training_instances	#positive_training_instances
Contract Title / Type	0,054%	4,157	3,64	261,003	15,133
Contracting Parties	0,105%	7,196	4,09	420,922	29,420
Start Date	0,030%	2,377	4,08	282,353	9,704
Effective Date	0,006%	631	4,34	91,244	2,743
Termination Date	0,004%	470	3,98	86,622	1,871
Contract Period	0,001%	355	3,88	64,611	1,378
Contract Value	0,017%	877	2,62	150,619	2,302
Governing Law	0,050%	2,155	6,05	369,839	13,052
Jurisdiction	0,038%	1,282	7,66	178,704	9,827
Legislation ref.s	0,412%	5,228	5,24	780,147	27,408
Clauses	-	~40K	-	~3M	~165K
Introduction	-	904	-	3,556	904

Legend of Table 6.2:

percentage: the percentage of tokens from the specified category against all words.
#elements: the number of annotated elements (multi-token phrases).
tokens per element: the average size of the element (multi-token phrases) in tokens.
#training_instances: the number of samples (tokens), which used for training.
#positive_training_instances: the number of annotated samples (tokens) for the specified category, which used for training.

6.2 Experimental Results

6.2.1 Training & Results

As was already mentioned in subsection 4.4.2 we performed experiments with Logistic Regression and Support Vector Machines algorithms using the implementations of the Scikit-learn[1] library [18]. In both cases, classifiers (estimators) were tuned via grid-search on the available parameters for each classifier. Except the library's available parameters for the algorithms, in our model there is one more essential hyper-parameter. As we already see in Table 6.2 the percentage of tokens for any specified category against all words is less than 1%. In order to handle the inbalance between positive instances and the rest of the negative ones, we need to sub-sample. So we specify an according hyper-parameter to tune the number of characters around the positive instances, we would like to consider in each experiment.

With respect to our word2vec model and model's data representation, we reach the outcomes presented on Table 6.3.

Table 6.3: Classification Experiments: Evaluation per Token

Algorithm	Logistic Regression			Support Vector Machines		
Category	precison	recall	F1	precision	recall	F1
Contract Title / Type	0,67	**0,80**	0,73	**0,84**	0,78	**0,80**
Contracting Parties	0,65	**0,89**	0,75	0,73	0,87	**0,79**
Start Date	**0,84**	0,94	0,88	0,82	**0,97**	**0,89**
Effective Date	0,76	0,80	0,78	**0,78**	**0,82**	**0,80**
Termination Date	0,75	0,88	0,81	**0,76**	**0,93**	**0,84**
Contract Period	**0,80**	0,60	0,69	0,78	**0,63**	**0,70**
Contract Value	**0,69**	0,46	0,55	0,66	**0,63**	**0,64**
Governing Law	0,70	**0,96**	0,81	**0,81**	0,94	**0,87**
Jurisdiction	0,56	**0,96**	0,71	**0,71**	0,85	**0,79**
Legislation ref.s	0,95	0,86	**0,90**	**0,95**	**0,86**	0,90
Clauses	-	-	-	**0,90**	**0,87**	**0,88**
Introduction	-	-	-	**0,91**	**0,90**	**0,90**

For the majority of categories, Support Vector Machines outperform Logistic Regression. In Figures 6.3-6.13, we present the learning curves for each category. A learning curve shows the validation and training score of a classifier for varying numbers of training instances. It is useful to find out the likelihood for the benefit we could have from adding more training data. Also we can observe, if the classifier suffers more from a variance or a bias error. If both the validation score and the training score converge to a specific value that is too low with increasing size of the training set, we will not benefit much from more training data. We will probably have to use another model or a parameterization of the current model that can learn more complex concepts (i.e. has a lower bias). If the training score is much greater than the validation score for the maximum number of training samples, adding more training samples will most likely increase generalization. There is no evidence that the results, in most categories except Contracting Parties and Termination Date, can be improved with further training of the same linear model.

[1] http://scikit-learn.org/

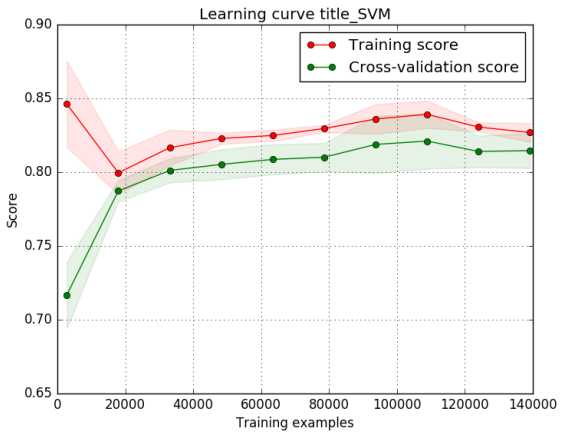


Figure 6.3: Contract Title / Type

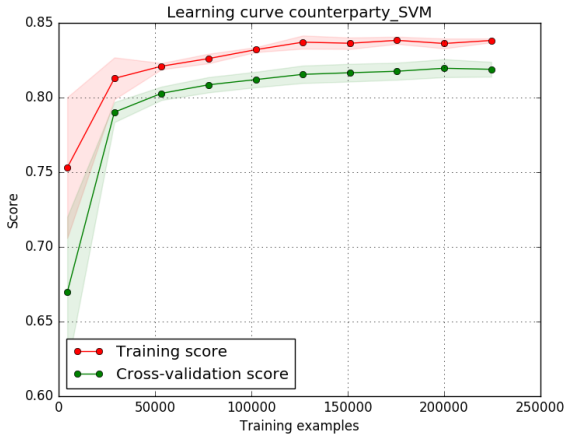


Figure 6.4: Contracting Parties

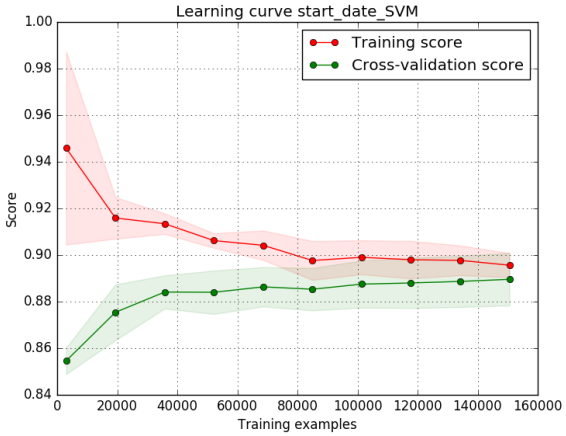


Figure 6.5: Start Date

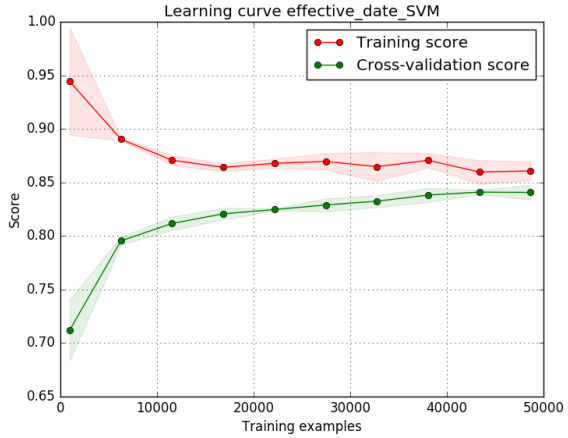


Figure 6.6: Effective Date

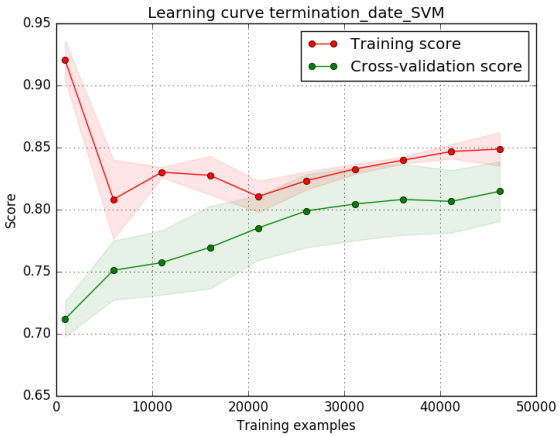


Figure 6.7: Termination Date

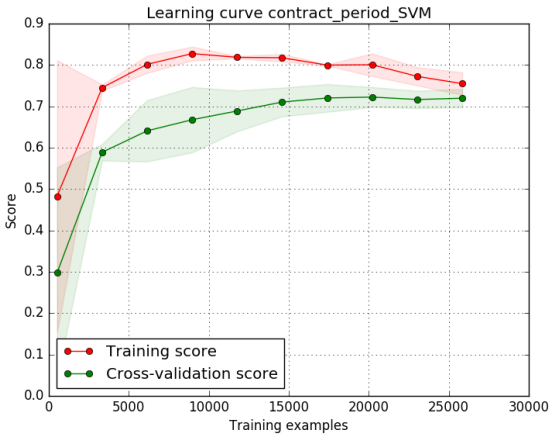


Figure 6.8: Contract Period

Figure 6.9: Learning curves using SVM training (Part 1)

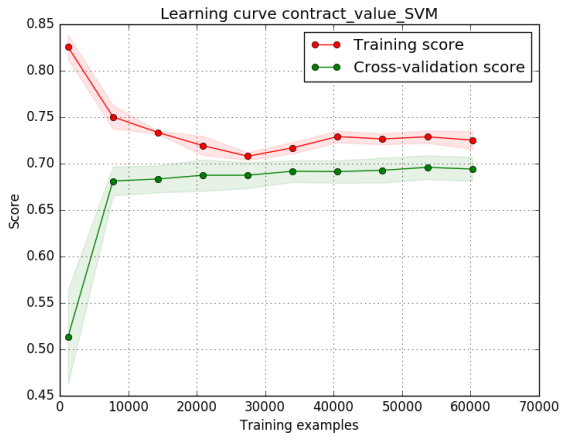


Figure 6.10: Contract Value

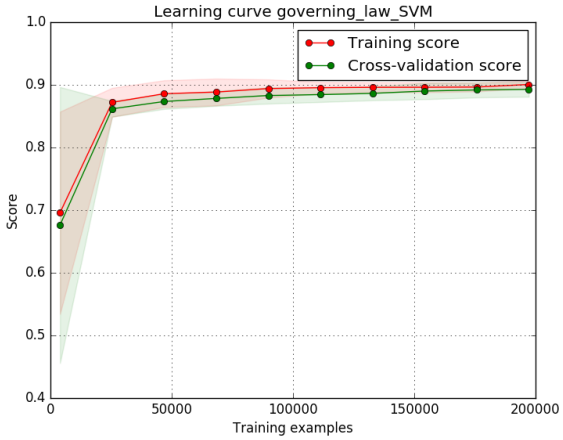


Figure 6.11: Governing Law

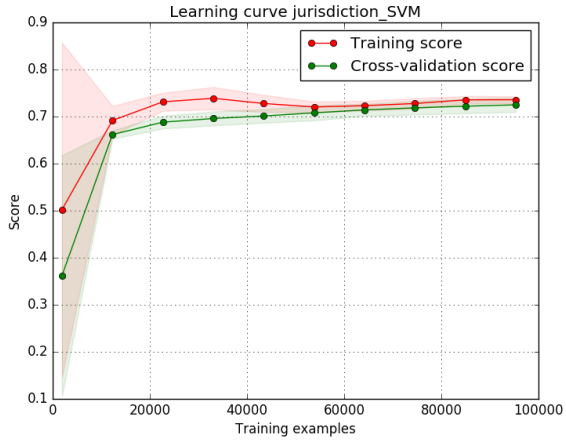


Figure 6.12: Jurisdiction

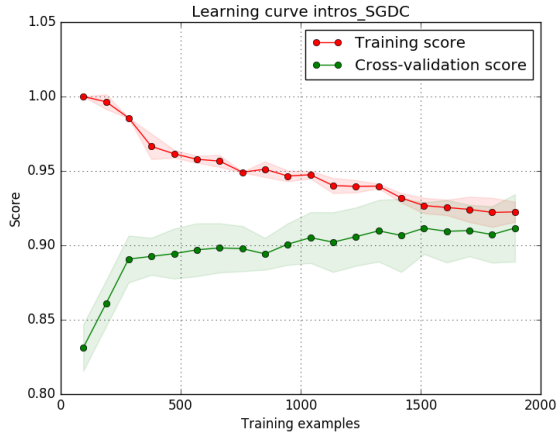


Figure 6.13: Introduction

Figure 6.14: Learning curves using by SVM training (Part 2)

In our initial assumption (See Section 1.1), we mentioned that *"the use of word embeddings will bring extra reliability against hand-crafted feature learning"*. In order to explore this assumption, we have to provide the necessary comparative experiments (See Table 6.4 and Figure 6.15). We set-up those experiments in four different configurations: (Model 1) will consider only word embeddings; (Model 2) will consider both word embeddings and the basic set of hand-crafted features; (Model 3) will consider only the full set of hand-crafted features and finally comes (Model 4) configuration considering both the word embeddings and the full set of hand-crafted features. Out of this scope, the experiments for introductions are performed per paragraph using a feature representation of tf-idf scores (Model 0).

Table 6.4: Classification Experiments: Evaluation per Token

Models	Model 1			Model 2			Model 3			Model 4		
Category	precision	recall	F1	precision	recall	F1	precision	recall	F1	precision	recall	F1
Contract Title / Type	0.78	0.69	0.73	0.64	0.83	0.72	**0.85**	0.74	0.79	0.84	**0.78**	**0.80**
Contracting Parties	0.42	**0.98**	0.59	0.49	0.95	0.65	**0.77**	0.76	0.76	0.73	0.87	**0.79**
Start Date	0.64	**0.99**	0.78	0.74	**0.99**	0.84	**0.82**	0.96	0.88	**0.82**	0.97	**0.89**
Effective Date	**0.79**	0.48	0.60	0.64	**0.91**	0.75	0.78	0.79	0.78	0,78	0,82	**0,80**
Termination Date	-	-	-	0.59	**0.98**	0.73	0.63	0.86	0.73	**0,76**	0,93	**0,84**
Contract Period	-	-	-	**0.96**	0.07	0.14	0.78	0.56	0.65	0,78	**0,63**	**0,70**
Contract Value	**0.74**	0.27	0.40	0.46	**0.71**	0.56	0.74	0.41	0.53	0,66	0,63	**0,64**
Governing Law	0.61	0.98	0.75	0.34	**0.99**	0.51	**0.89**	0.85	**0.87**	0,81	0,94	**0,87**
Jurisdiction	0.51	0.94	0.66	0.40	**0.96**	0.56	**0.83**	0.63	0.72	0,71	0,85	**0,79**
Legislation ref.s	0.52	**0.91**	0.66	0.77	0.86	0.82	0.94	0.85	0.89	**0,95**	0,86	**0,90**
Clauses	0.08	**0.99**	0.15	0.56	0.91	0.70	0.77	0.87	0.82	**0,90**	0,87	**0,88**
Models										Model 0		
Introduction										**0,91**	**0,90**	**0,90**

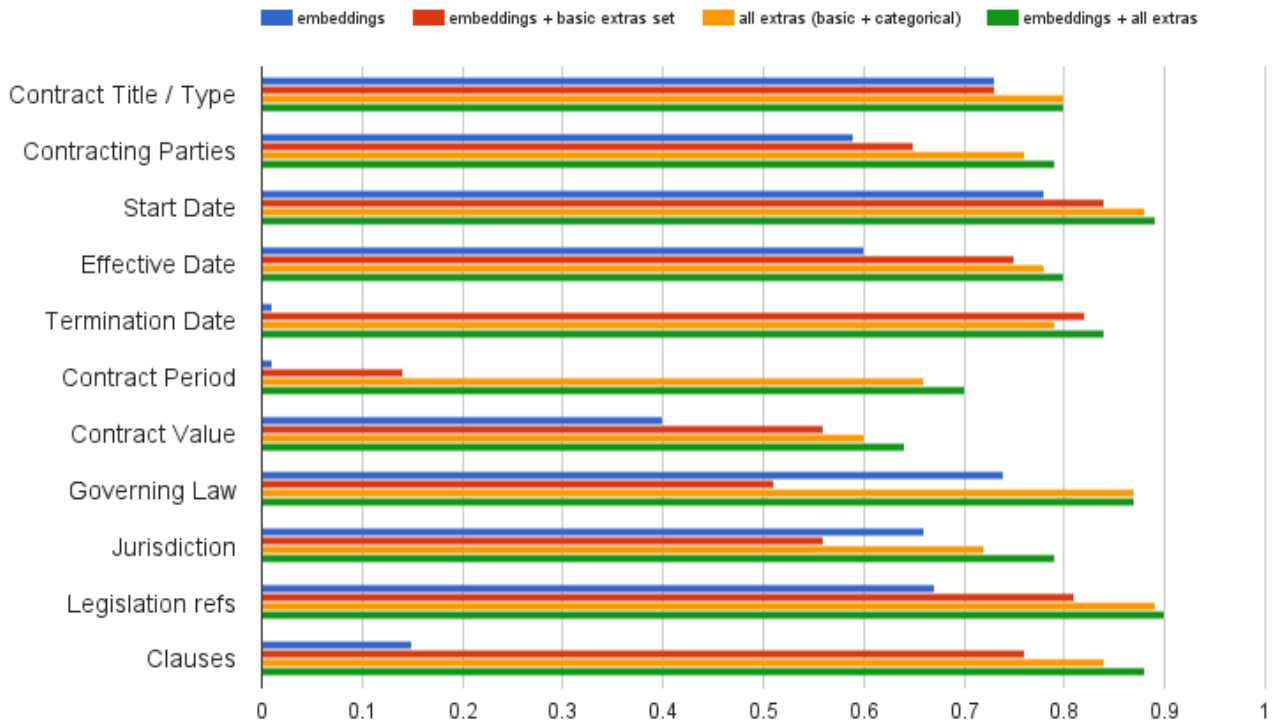


Figure 6.15: Classification Experiments: Embeddings - Hand-crafted features Impact (F-score)

In terms of data representation we need to highlight that our domain-specific word embeddings (Model 1) perform decently on their own, excluding the categories which are supported with a few samples and consequently cannot perform sufficiently (See Table 6.2. The performance was greater by giving the basic set of features (Model 2), which provide a vital boost, in some categories, capturing basically case sensitivity. Considering both word embeddings and hand-crafted features (Model 4), we produce the best results. We strongly believe that the corresponding results are highly relevant with the quality of the word embeddings. This belief is not only related with the domain-specific aspect, but also with the actual training of the word2vec model in terms of the corpora size. For this purpose we propose to perform training in a much larger corpus with hundred of thousands contracts.

6.2.2 Post-processing & Results

The experiments' evaluation gives us high expectations for applying our model in real-case scenarios. In this project we make one leap forward, from academic experimentation to the real-world applications. Limiting our model's evaluation through per-token measurements, completely misleads our objective. We want the final outcomes of our system to be multi-token groups (phrases) which comply to the actual elements (entities). Our study leads to the conclusion that there cannot be reliable results without extensive use of post-processing upon classifier models.The post-processing task not only sustains the classification standards but also improves them (See Table 6.5).

Table 6.5: Post-processing task: Evaluation per Element

Algorithm	Logistic Regression			Support Vector Machines		
Category	precison	recall	F1	precision	recall	F1
Contract Title / Type	0,83	0,83	0,83	**0,85**	**0,85**	**0,85**
Contracting Parties	0,79	**0,84**	**0,81**	**0,81**	0,82	**0,82**
Start Date	**0,92**	0,91	0,91	**0,92**	**0,93**	**0,92**
Effective Date	0,91	0,86	0,88	**0,93**	**0,87**	**0,90**
Termination Date	0,88	0,88	0,88	**0,90**	**0,90**	**0,90**
Contract Period	**0,65**	0,67	0,66	**0,65**	**0,73**	**0,69**
Contract Value	**0,86**	0,80	**0,83**	0,85	**0,82**	**0,83**
Governing Law	0,88	0,91	0,89	**0,91**	**0,92**	**0,92**
Jurisdiction	0,82	**0,78**	0,80	**0,91**	**0,78**	**0,83**
Legislation ref.s	**0,95**	**0,97**	**0,96**	0,95	0,95	0,95
Clauses	-	-	-	**0,94**	**0,95**	**0,95**

Bypassing such a critical process our system totally collapses in most cases (See Table 6.6).

Table 6.6: Post-processing task: Post-processing Impact (F-score)

Approach	ML			ML + Rules		
Category	precision	recall	F1	precision	recall	F1
Contract Title / Type	0.83	0.77	0.80	**0,85**	**0,85**	**0,85**
Contracting Parties	0.59	0.80	0.68	**0.81**	**0,82**	**0,82**
Start Date	0.70	0.93	0.80	**0.92**	**0.93**	**0.92**
Effective Date	0.67	0.66	0.66	**0.93**	**0.87**	**0.90**
Termination Date	0.75	0.75	0.75	**0.90**	**0.90**	**0.90**
Contract Period	0.32	0.49	0.39	**0.65**	**0.73**	**0.69**
Contract Value	0.78	0.75	0.76	**0.85**	**0.82**	**0,83**
Governing Law	0.55	0.89	0.68	**0.91**	**0.92**	**0.92**
Jurisdiction	0.38	0.74	0.50	**0.91**	**0.78**	**0.83**
Legislation ref.s	0.67	0.81	0.73	**0,95**	**0,97**	**0,96**
Clauses	0.24	0.72	0.36	**0,94**	**0,95**	**0,95**

6.2.3 Our system vs Rule-based approaches

In order to have a good perception of the respective baselines and the comparative perspective between them and our own system we built some advanced rule-based entity recognisers for each category, based on our domain knowledge and linguistic analysis. In most cases, the rules are applied in two sub-tasks. For the first sub-task of recognition there are two approaches: the application of regular expresions capturing the pattern of the specified element; or the sequence tagging per token based on bags of words. The second sub-task is the same validation process, which we used during the post-processing in our model.

For a better understanding, we present two examples for the rule-based hand-crafted entity recognizers:

Legislation References (Regular Expression)

(a) Find word sequences starting with as many words with at least the first letter in uppercase or specific stop words (i.e. and, the, of); followed by one of the words: Act, Code, Regulation(s) or Ammendment(s); followed by a year expression in a four-digit fashion.

(b) Eliminate possible stop words in the beggining of the word sequence.

Effective Date (Sequence Tagging)

(a) Iterate over all words and label those words which are included in the general dates' bag of words or it is a comma (,).

(b) Group sequences of positively labeled words.

(c) Examine if the 5 previous or the 5 next words around the sequence are one of ”effective” or ”effect”.

This whole process is necessary in order to have a good perception of the respective baselines and the comparative perspective between them and our own system ((See Table 6.7, Figure 6.16).

Table 6.7: Baseline Comparison: Rules vs ML + Rules

Approach	Rules			ML + Rules		
Category	precision	recall	F1	precision	recall	F1
Contract Title / Type	**0.92**	0.80	**0.85**	0,85	**0,85**	**0,85**
Contracting Parties	0.78	0.43	0.55	**0.81**	0,82	**0,82**
Start Date	0.87	0.91	0.89	**0.92**	**0.93**	**0.92**
Effective Date	**0.93**	**0.87**	**0.90**	**0.93**	**0.87**	**0.90**
Termination Date	**0.94**	0.87	**0.90**	0.90	**0.90**	**0.90**
Contract Period	0.22	**0.75**	0.34	**0.65**	0.73	**0.69**
Contract Value	0.66	**0.92**	0.77	**0.85**	0.82	**0,83**
Governing Law	**0.92**	0.88	0.90	0.91	**0.92**	**0.92**
Jurisdiction	0.88	0.53	0.66	**0.91**	**0.78**	**0.83**
Legislation ref.s	**0.98**	0.94	**0.96**	**0,95**	**0,97**	**0,96**
Clauses	0.90	0.94	0.92	**0,94**	**0,95**	**0,95**
Introduction	0.88	**0.93**	**0.90**	**0,91**	0,90	**0,90**

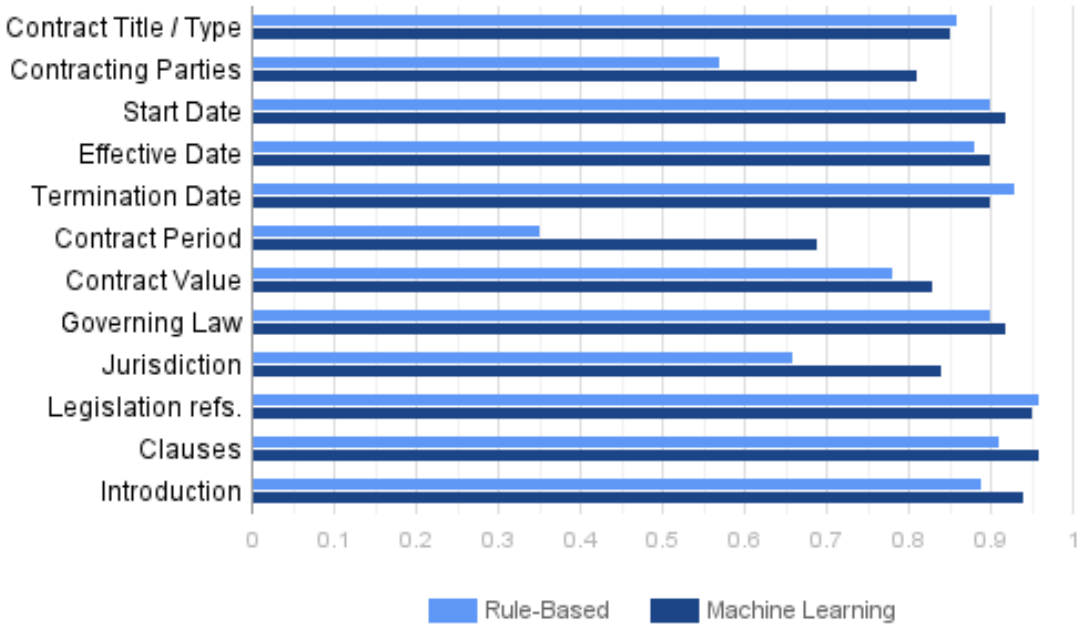


Figure 6.16: Baseline Comparison: Rule-based vs Machine Learning (F-score)

There are some interesting observations:

(a) In elements with a very consistent pattern (i.e. legislation, dates), rule-based approaches are really competitive with our system.

(b) In elements with inconsistent shape (i.e. contracting parties, jurisdiction), our system outperforms rule-based approaches.

(c) In some elements (dates, contract value) regular expression patterns give high recall rate, but thrusting down on precision, unable to capture the context around the elements and distinct similar cases, a phenomenon which seems to be handled by our models.

(d) There are some models which seem to get poor results due to insufficient training (See Table 6.2).

7. CONCLUSION AND FUTURE WORK

7.1 Conclusions

Our initial assumption (See Section 1.1) was that *"the use of word embeddings will bring extra reliability against both hand-crafted feature learning"*, a point which successfully proved through extensive experimentation and comparative evaluations (See subsection 6.2.1). Word embeddings seem to provide the demanded extra reliability on some exceptional cases that hand-crafted features are unable to handle. We have to notice that domain-specific word embeddings perform much better than general purpose embeddings.

Our persistence for the necessity of post-processing also proved important in order to provide a usable system for real-case scenarios. The evaluation experiments show that the plain use of the classifiers is not possible for production(See subsection 6.2.2).

The comparative experiments between our system and rule-based hand-crafted approaches show that in most cases, even such a simple machine learning approach performs better than the baseline rule-based systems and give us high expectations to continue our research with more challenging approaches in the future (See section 7.2) to totally outperform rule-based ones.

This thesis can be used as a road-map for the implementation of such systems from the early stage of data collection to the very end of production.

7.2 Future Work

Domain-specific word embeddings - Basic hand-crafted features embeddings

As already mentioned above (See subsection 6.1.1), we believe that there is great potential to create more accurate and representative word embeddings by using a massive corpus with hundreds of thousands of contracts and performing the appropriate tuning. It also makes great sense to build embeddings for our basic hand-crafted features (i.e. case-sensitivity) and posibly use some new features, like part-of-speech (POS) tags embeddings. We are going to proceed in this direction to deepen the transition from the partially sparse vector space into a well-structured dense one.

Multi Layer Perceptron (MLP)

Neural Networks are consistently gaining more research interest from our community. They are considered to be the state-of-the-art learning models and manage to obtain improved results compared with the traditional linear models. In our objective to improve the performance of our system, will proceed in learning using MLP models.

Recurrent Neural Networks (RNNs)

Discussing neural networks, Recurrent Neural Networks (RNNs) seem to have excellent results for language modelling and is a must case-study for our future experiments. Bi-directional RNNs [19] could be used as a similar approach with ours (windows of tokens), approximating both past and future context. We believe this technique is highly promising in our case.

LIST OF ABBREVIATIONS

ERP	Enterprise Resource Planning
NER	Named Entity Recognition
POS	Part-Of-Speech
NLP	Natural Language Processing
SVM	Support Vector Machines
UI	User Interface
MLP	Multi Layer Perceptron
RNN	Recurrent Neural Networks
HTTP	HyperText Transfer Protocol
CSS	Cascading Style Sheets
JS	Javascript

LIST OF TRANSLATIONS

NLP	Επεξεργασία Φυσικής Γλώσσας
Word Embeddings	Διανυσματικές Παραστάσεις Λέξεων

APPENDIX A. EXPERIMENTS' SET-UP

A.1 Software Used in the Experiments

The code base for the experiments was built upon the following technologies:

Programming Language:	Python 3.5[2]
Machine Learning Library:	scikit-learn 0.17.1[3]
NLP Tools (Tokenizer):	nltk 3.2.1[4]
Word2vec Support:	gensim 0.13.2[5]

A.2 Parameters of Experiments

The hyper-parameter values that were used to produce word2vec embeddings are presented in Table A.1.

Table A.1: Word2Vec Parameters

Parameter	Value
Algorithm	Skip-Gram
Negative-Sampling	5
Vector size	200
Iterations / Epochs	10
Minimum occurences / token	5

During classification experiments, multiple values were selected for the available hyper-parameters. The final selection is presented in Table A.2.

Table A.2: Hyper-Parameters of Classification Experiments

Hyper Parameters	
Parameter	Value
Sample Padding	500 characters
Window Size	11 tokens
Classification Parameters	
Logistic Regression[1]	
Parameter	Value
Penalty	l2
Max Iteration	1000
Warm Start	True
C	0,001
SVM[2]	
Parameter	Value
Loss	Hinge
Penalty	l2
Max Iterations	1000
C	0,001
SVM (SGD)[3]	
Parameter	Value
Loss	Hinge
Penalty	l2
Max Iterations	1000
Warm Start	True
Alpha	0,001

REFERENCES

[1] Michael Curtotti and Eric Mccreath. Corpus Based Classification of Text in Australian Contracts. In *Proceedings of the Australasian Language Technology Association Workshop 2010*, pages 18–26, Melbourne, Australia, 2010.

[2] Christopher Dozier, Ravikumar Kondadadi, Marc Light, Arun Vachher, Sriharsha Veeramachaneni, and Ramdev Wudali. Named Entity Recognition and Resolution in Legal Text. *Semantic Processing of Legal Texts*, pages 27–43, 2010.

[3] Paulo Quaresma and Teresa Gonçalves. Using Linguistic Information and Machine Learning Techniques to Identify Entities from Juridical Documents. *Semantic Processing of Legal Texts*, pages 44–59, 2010.

[4] Enrico Francesconi and International Conference on Language Resources & Evaluation International Workshop on Semantic Processing of Legal Texts. *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*. Springer, 2010.

[5] Zia Ahmed and Rajkiran Veluri. Named Entity Recognition and Question Answering Using Word Vectors and Clustering Problem Statement. *CS229: Machine Learning Project, Computer Science Department, Stanford University*, 2015.

[6] Cícero Nogueira dos Santos and Victor Guimarães. Boosting Named Entity Recognition with Neural Character Embeddings. *CoRR*, abs/1505.05008, 2015.

[7] Katharina Scharolta and Sienčnik. Adapting word2vec to Named Entity Recognition. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015*, pages 239–243, Vilnius, Lithuania, 2015.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. *CoRR*, abs/1310.4546, 2013.

[10] Jonathon Shlens. A Tutorial on Principal Component Analysis. *CoRR*, abs/1404.1100, 2014.

[11] Thomas Asikis. Part of Speech Tagging in Greek Texts using Word Embeddings and Deep Neural Networks. MSc thesis (in Greek), Department of Informatics, Athens University of Economics and Business, 2016.

[12] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, 37:1554–1563, 1966.

[13] Richard Socher, Cliff C. Lin, Andrew Y. Ng, and Christopher D. Manning. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, Montreal, Canada, 2011.

[14] David R. Cox. The regression analysis of binary sequences (with discussion). *J Roy Stat Soc B*, 20:215–242, 1958.

[15] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[16] Christopher J.C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[17] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, 2010.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Mike Schuster and Kuldip K. Paliwal. Bidirectional Recurrent Neural Networks. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 45(11):2673–2681, 1997.